Distributed, Adaptive Control of Advanced Life Support Systems

David Kortenkamp

NASA Johnson Space Center/Metrica Inc.

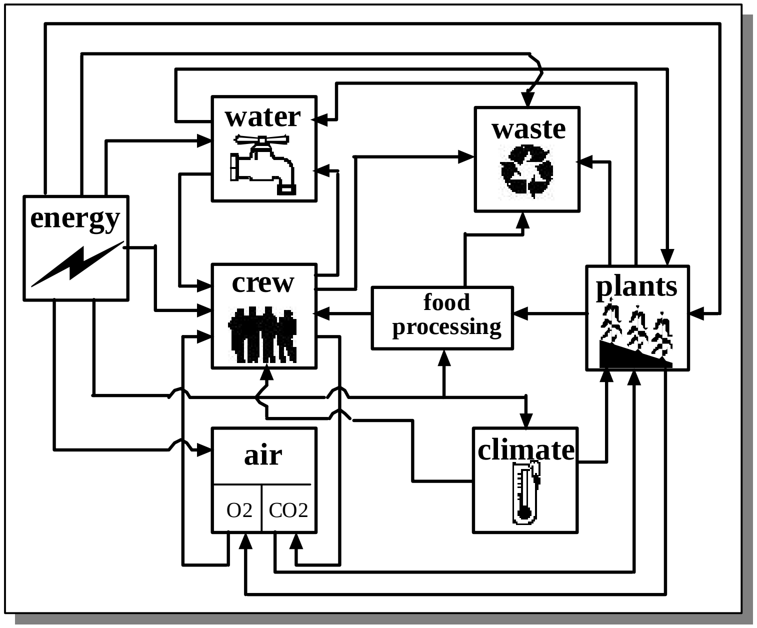
Houston TX 77058

kortenkamp@jsc.nasa.gov

http://www.traclabs.com/~korten

Advanced Life Support Systems

- Regenerative
  - produce own food
  - recycle water and air
- Low margins, volume, mass, energy and labor
- Limited resupply
- Highly interconnected
- Require optimization and tight control



David Kortenkamp
NASA Johnson Space Center

Control Issues

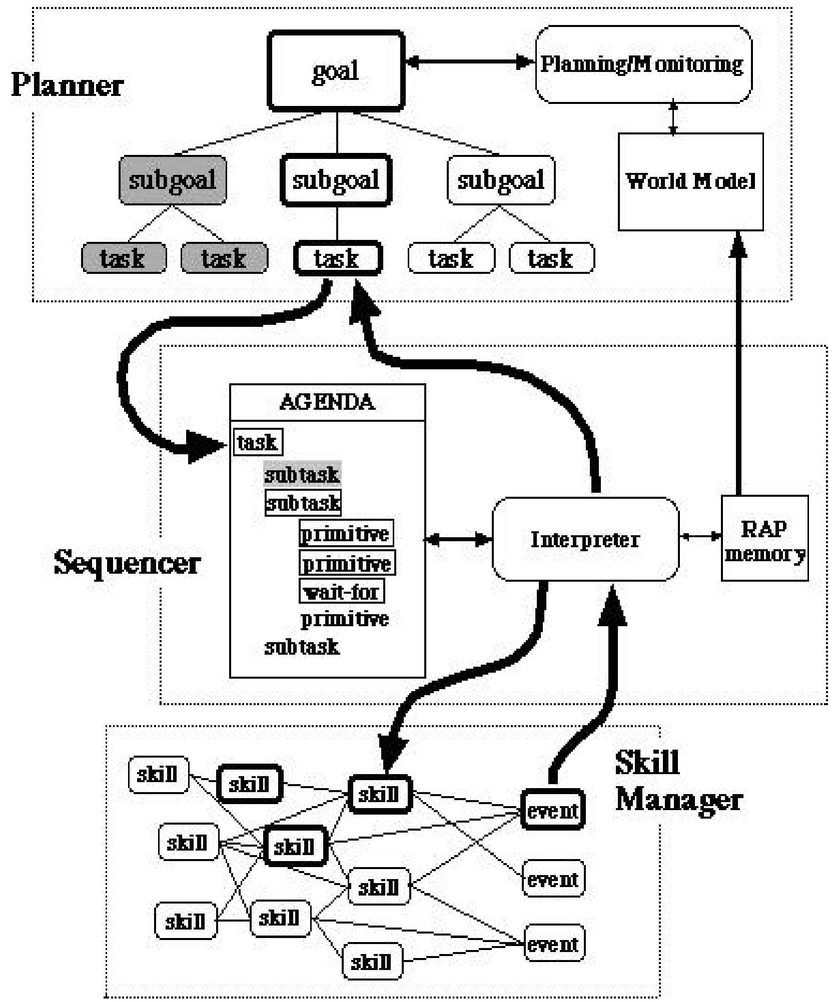
- Advanced Life Support (ALS) systems are:
  - *Dynamic* – it is not sufficient only to find a single *a priori* setting
  - *Non-stationary* – presence of adaptive organisms such as humans, plants and bacteria as well as degradation requires adaptation
  - *Safety-sensitive* – crew depends on system for life support, verification and validation are important

Verification and Validation Issues

- Advanced Life Support Systems pose unique V&V challenges
  - Control system needs to detect trends towards failure, which may be months away, in addition to simple failures
  - Verification of models/simulations
  - Modeling of biological processes
  - Distributed control has its own set of V&V issues, especially with respect to timing and communication
- Work-to-date has been in a research setting, so no formal V&V procedures

Previous and Current Control Systems

- Several experiments at JSC based on the 3T control architecture
- 3T
  - planning
  - sequencing
  - control



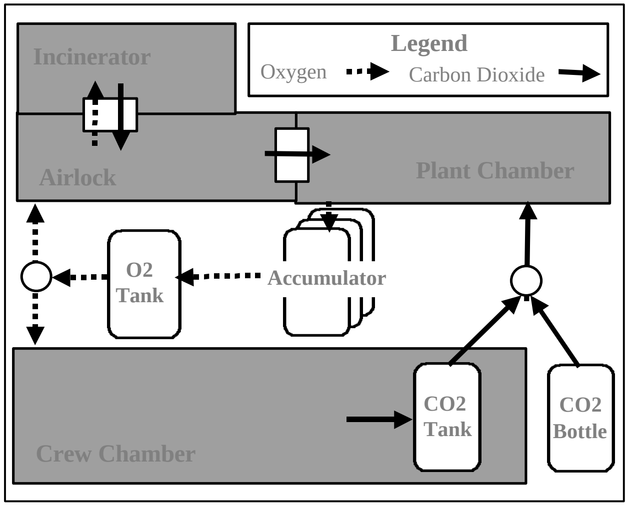
David Kortenkamp
NASA Johnson Space Center

Overall V&V Strategy

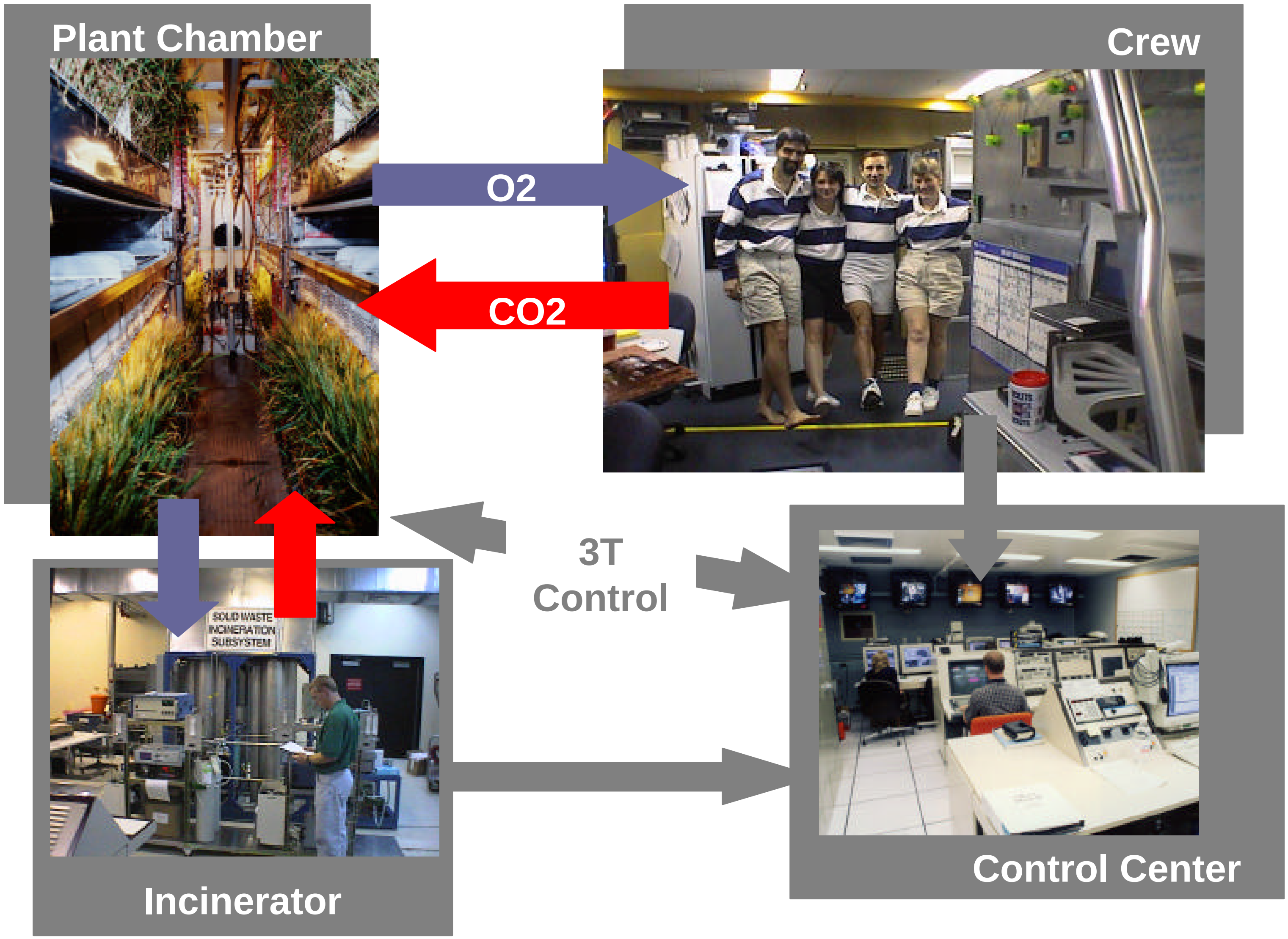
- First integrate the lowest tier (skills) with hardware; manually activate low level controls through computer
- Next integrate the middle tier (sequencer) with skills controlling hardware; manual activation of sequences
- Finally, integrate the top tier (planner) with the integrated sequences and skills
- At end of each phase, there is a usable control capability
- Separate skill managers and RAPs for separate control entities

Phase III Crewed Test

- Four crew members for 91 days in a closed chamber
- Wheat crop in another chamber
- 3T managed transfer of gases between the two chambers
- Operated reliably round-the-clock for 73 days (10/6-12/19)
- Typically ran without human supervision or intervention



David Kortenkamp
NASA Johnson Space Center



David Kortenkamp
NASA Johnson Space Center

Verification and Validation

- Phase I
  - Demonstrate feasibility
  - 5 months development, 3.5 months testing (2 programmers)
- Following steps performed in order
  - Stand-alone testing
    - RAPs with emulated skills in lab
    - Skills via user activation in lab
    - Skills with recorded data
  - Interface testing
    - Test data interface between life support DAQ and skills
    - Test RAPS to skills interface using actual data (monitor only)

Verification and Validation (cont.)

- Phase 1 continued
  - Integrated testing
    - Integrated test of RAPs and skills with hardware during wheat test (advisory only)
    - Integrated test of RAPs and skills with hardware during wheat test (control, only during workday, then 24 hours)

Verification and Validation (cont.)

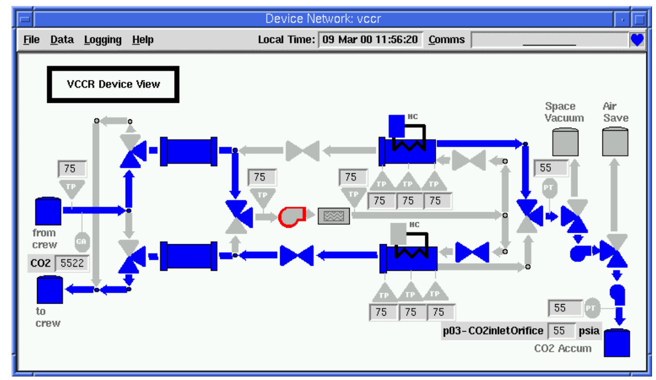
- Phase II
  - Deploy operational system
  - 2 months development, 2 months integration and testing (2 programmers)
- Following steps performed in order
  - Stand-alone testing
    - Emulated skills
    - Test skills with user activation
    - Test skills with recorded data
  - Interface testing
    - Test interface between DAQ and skills
    - Test interface between RAPs and skills
    - Test interface between planner, RAPs and emulated skills

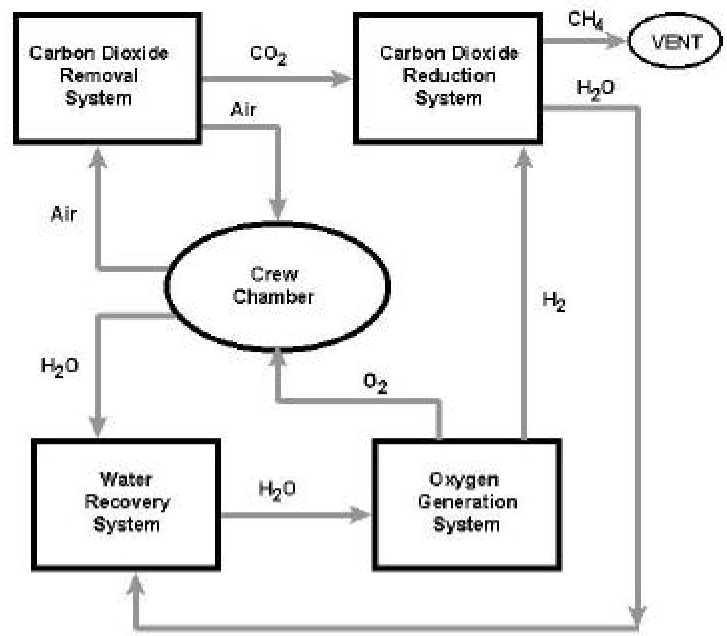
Verification and Validation (cont.)

- Integrated testing
  - RAPs and skills with CONFIG simulation
    - Checked out standard operating procedures
    - Checked out fault management procedures
  - RAPs and skills with actual hardware
  - Planner, sequencer and skills with CONFIG simulation
- Operational testing
  - RAPs and skills for 73 days
  - Planner, sequencer and skills near end of test
- Data recorded including commands and sensors
- See Schreckenghost *et al* IAAI-98

Air Revitalization System

- Simulation of an ARS using a discrete event simulator (CONFIG)
- 3T control integrated with Livingstone MIR (from Ames)
- Planning currently being added





David Kortenkamp
NASA Johnson Space Center

Verification and Validation

- Stand-alone testing for code validation
  - Skills with user activation
  - Livingstone models with scripts emulating data messages from skills
  - RAPs with emulated skills
  - Planner with sequencer and emulated skills
- Interface testing
  - Test Livingstone interfaces with emulated skills
  - Test interface between CONFIG and skills

Verification and Validation (cont.)

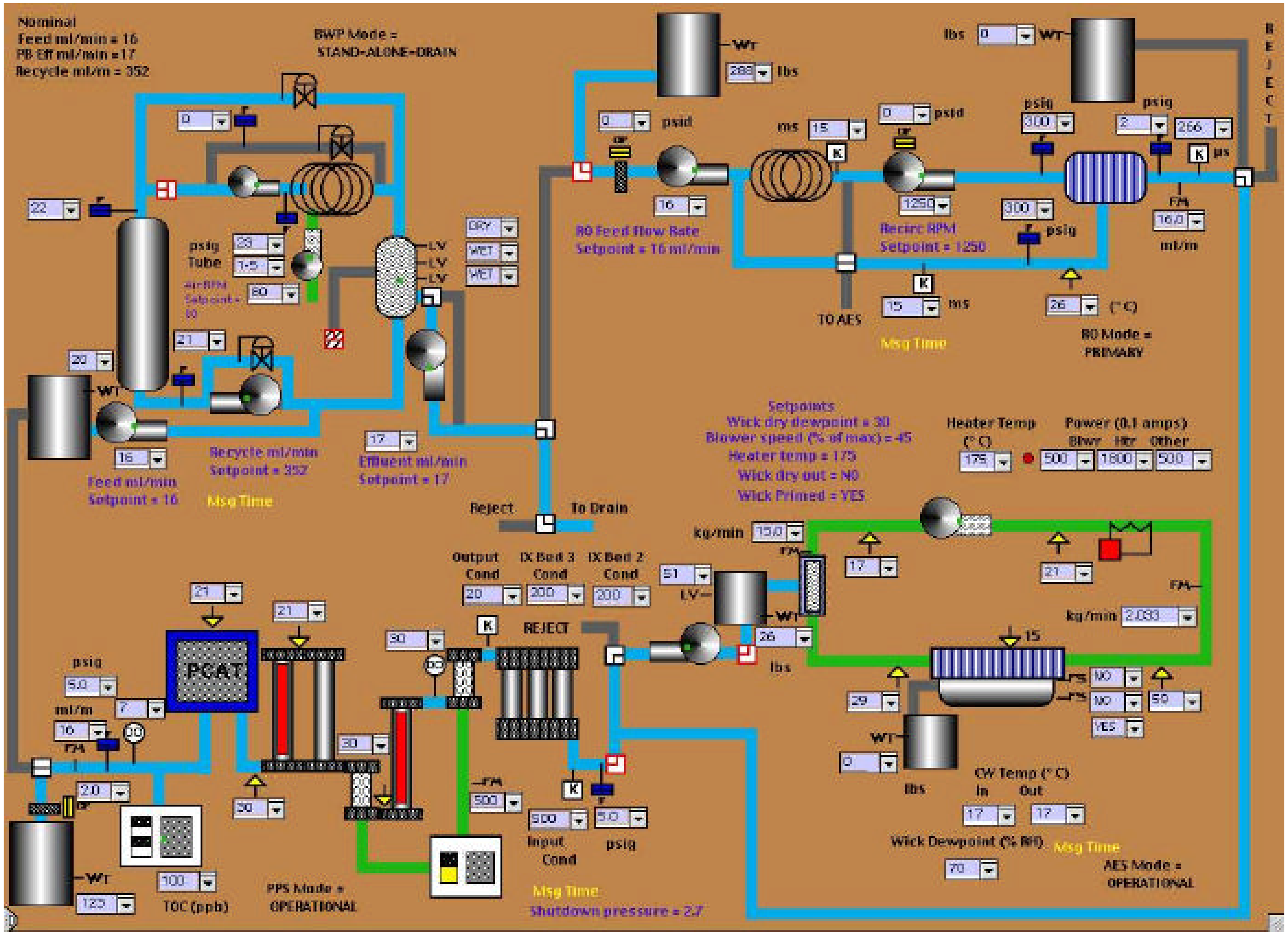
- Integration testing
  - Integrated testing with RAPs and skills using CONFIG simulation
  - Integrated test of Livingstone, RAPs and skills with CONFIG
  - Integrated test of planner, Livingstone, RAPs and skills with CONFIG
- See Malin *et al* IEEE Aerospace Conference 2000

Water Recovery System

- Four integrated subsystems:
  - Biological water processor (BWP)
  - Reverse osmosis system (RO)
  - Air evaporation system (AES)
  - Post-processor (PPS)
- 3T skills (over 75 separate skills)
- 3T RAPs
- ~200 sensors and actuators

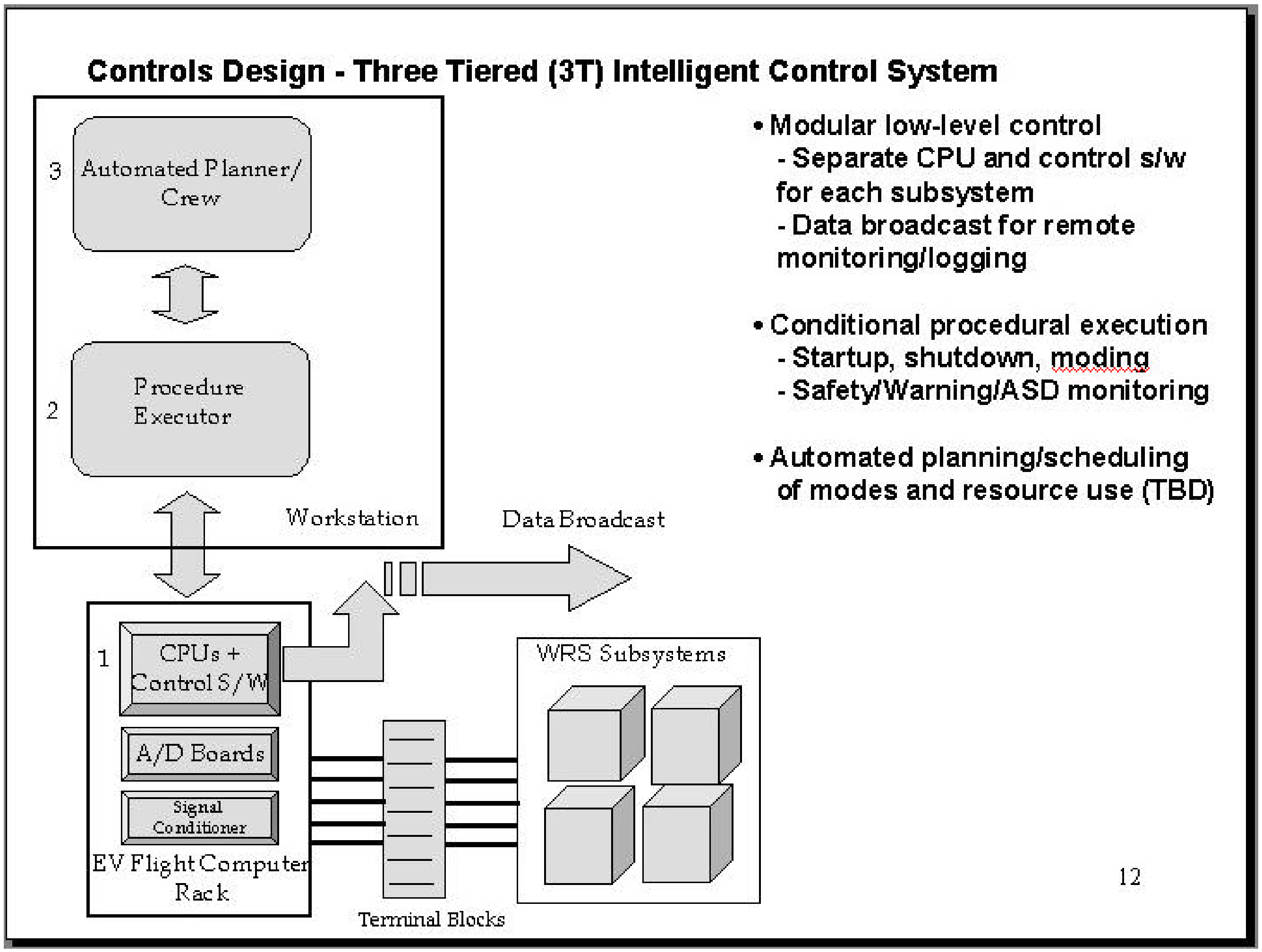
David Kortenkamp
NASA Johnson Space Center

Integration of Subsystems



David Kortenkamp
NASA Johnson Space Center

Control Strategy



David Kortenkamp
NASA Johnson Space Center

Verification and Validation

- Get sequence and state diagrams from design engineers
- Implement code and test with simple simulation
- With hardware, do the following
  - Each subsystem standing alone
    - Calibrate all instruments through the skills
    - Test/verify each low-level RAPs query and action
    - Test/verify each mid-level RAPS
    - Test/verify high-level RAPS
  - Integration tests (using de-ionized water)
    - Test BWP+RO through all test points
    - Test RO+AES+PPS through all test points
    - Test all four systems through each test point
  - Duration: Conduct integration tests for 72 hours

Verification and Validation (cont.)

- Actual tests (using mix of urine and waste water)
  - Record all data. Analyze off-line daily. Correct control anomalies as necessary.
  - For code changes during test, retest only those portions affected as determined by code inspection and simulation
- Key is that subsystems are treated as independent agents (horizontal modularity) and the layered control gives us vertical modularity
- From Pete Bonasso, NASA JSC/Metrica Inc.

Lessons Learned

- Routine control of complex life support systems is within reach
- Small changes to sensor calibration or the underlying biological/chemical processes requires expensive recoding of control procedures
- Changes to the desired operating regime (e.g., optimizing for a different resource) requires expensive recoding of control procedures
- Complex interactions are difficult to predict
- Adaptation of control code is required for long-duration, autonomous missions

Learning in ALS Systems

- Most of the control will be hand-coded and fixed
- Some portion will need to adjust as the system runs
- Many open research questions
  - On-line vs. off-line learning
  - Experimentation with the real system
  - Fidelity of models and relationship to learning
  - Abstraction of state and action space (making system aware of hidden states)
  - Crew interaction with learning system (inspectability and instructability)

The Role of Learning

- Detecting signatures
  - Parsing real-world data stream to recognize events
- Refining models
  - Using feedback from actual system to adjust models
- Robust design
  - Searching through design criteria for optimal solution
- Learning/optimizing sequences
- Integrating with autonomous control
- Adaptive crew interfaces
- Control system design methodology
  - Using learning algorithms to find important variables and interactions
  - Helps overcome some V&V issues since code is hand-written
- Optimizing resource allocation

ALS State Space

- Potential state space is enormous and hybrid (i.e., mix of discrete and continuous) so we need to abstract
- Possible abstractions are
  – Current levels of consumables (air, water, food)
  – Quality of air and water and health of plants
  – Flow paths for water and air through the system
  – Current energy allocations to subsystems
  – The current phase of operation
  – Crew health/happiness
  – Temperatures and other environmental measures

ALS Action Space

- Potential action space is large
- Combination of physical actions to produce abstract actions
  - Allocation of energy amongst subsystems
  - Use of consumable stores
  - Crew activity
  - Routing of air/water flows
  - Planting/harvesting of crops (when and which)
  - Adjusting crop light levels
  - Adjusting climate controls
  - Venting of gases to the outside atmosphere

ALS Rewards and Feedback

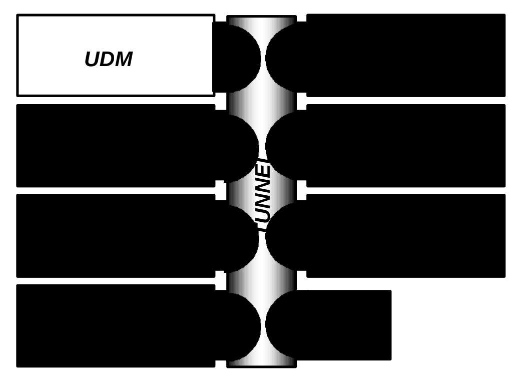
- Final or end state rewards
  – Duration of mission with different controllers
  – Total crew productivity over mission duration
  – Total amount of air, water, food or energy available in system or stores
- Progress measures
  – Quality of air and water and health of plants
  – Plant growth rates or plant food output
  – Climate feedback (keeping climate parameters within boundaries)
  – Health/satisfaction of crew

Future Work

- Begin working on other machine learning application areas
  - Sequence learning
    - Learning contexts as well as sequences
  - Integration with control systems
    - How good does initial control have to be for on-line learning to work?
    - How does control system decide when to devote resources to learning and when to use new knowledge?
- Investigate other ML techniques (memory-based, Samuel)
- Continue to explore theoretical issues of abstraction and model fidelity requirements
- Issue challenge to AI research community and make simulation available to all
- Begin applying techniques to real-world ALS testbeds

BIO-Plex

- Ground testbed of ALS system being built at NASA JSC
- Crew of 4 for up to 540 days
- 90% of food grown in chambers
- Testing starts in 2004
- Autonomous operation is the goal
- Testbed for future mission ops



David Kortenkamp
NASA Johnson Space Center

BIO-Plex challenges

- Dynamic system with thousands of sensors/actuators
- Need for mission planning/scheduling
- Modeling biological processes
- Sensor interpretation
- Natural crew interfaces
- 24/7 operation

ALSS Autonomy Roadmap

	FY01	FY02	FY03	FY04	FY05	FY06	FY07
Planning/Scheduling	Simple task planning for single subsystems	Planning for different time scales	Mixed-initiative and crew activity planning	Crop and menu planning	Continuous planning and replanning		
Executive	Single system procedures	Probabilistic reasoning about task context	Distributed, cooperating executives	Reasoning about procedure execution	Procedure synthesis		
Machine Learning	Parameter tuning	Learning/refining models from system data	Learning cross-system optimal control policies	Optimizing control in changing environments	Continuous learning		
Model-based Reasoning	Multi-step reconfiguration	Hybrid discrete/ continuous models	Hierarchies of models for reasoning across subsystems	Modeling and reasoning about software procedures	Procedure synthesis from models		
Sensor Interpretation	Sensor fusion	Automatic event recognition	Automatic sensor calibration	Interpretation of sensor nets	Vision for crop inspection and crew tracking		
Distributed Control	System architecture	Communication protocols and APIs for distributed components	FDIR on control system components	Dense networks of distributed sensors	Automated recovery from major control system failures		
Human Interaction	Natural language discourse with control system	Situation views of control system status	Mixed-initiative planning interfaces	Mobile computing for control system	Crew tracking and plan recognition		
Robotics	Autonomous control of Traybot	Planning and control of simulated robots	Shared control of EVA rovers and maintenance robots	Plant chamber automation for food processing	Test of BIO-Plex IVA maintenance and inspection robot		
Intelligent Data Understanding	Data models for storing data in database	Use of stored data to automatically refine models and simulations	Automated inventory control system	Automatic identification of significant events	Sophisticated analysis of trends and events		
Tests	ARS simulation WRS hardware	Preliminary Mission Simulation	120-day BIO-Plex test	240-day BIO-Plex test	240-day BIO-Plex test		540-day BIO-Plex test

David Kortenkamp
NASA Johnson Space Center

Conclusions

- BIO-Plex is not a flight system, so we can test advanced concepts in adaptive control and validation and verification
- Tightly coupled, real-time systems that are adaptive will require a different kind of V&V
- Want to move from systems that require extreme crew vigilance to systems that run on their own
  – Still need *adjustable autonomy* see tutorial by Dorais and Kortenkamp on my home page
- Developing a suite of simulations that we can distribute